**1.检查robots.txt**

大多数网站都会定义robots.txt文件，这样可以让爬虫了解爬去该网站时存在哪些限制。  
eg:<http://example.webscraping.com/robots.txt>

**2.识别网站所用技术**

pip install builtwith

import builtwith  
builtwith.parse('<http://example.webscraping.com>')  
{u'javascript-frameworks': [u'jQuery', u'Modernizr', u'jQuery UI'], u'web-framew  
orks': [u'Web2py', u'Twitter Bootstrap'], u'programming-languages': [u'Python'],  
u'web-servers': [u'Nginx']}  
使用了Python的web2py框架，另外还用了一些通用的JavaScript，因此该网站的内容很有可能嵌入在HTML中的，相比而言比较容易抓取。而如果改用AngularJS构建该网站，此时的网站内容就很用可能是动态加载的。另外，如果网站使用了ASP.NET，那么在爬取网页时，就必须要用到会话管理和表单提交了。

**3.寻找网站的所有者**

pip install python-whois

import whois  
print whois.whois('appspot.com')  
"country": "US",  
"whois\_server": "whois.markmonitor.com",  
"state": "CA",  
"registrar": "MarkMonitor, Inc.",  
"referral\_url": null,  
"address": "1600 Amphitheatre Parkway,",  
"name\_servers": [  
"NS1.GOOGLE.COM",  
"NS2.GOOGLE.COM",  
"NS3.GOOGLE.COM",  
"NS4.GOOGLE.COM",  
"ns4.google.com",  
"ns3.google.com",  
"ns1.google.com",  
"ns2.google.com"  
],  
"org": "Google LLC",  
"creation\_date": [  
"2005-03-10 02:27:55",  
"2005-03-09T18:27:55-0800"  
],  
"emails": [  
"[abusecomplaints@markmonitor.com](mailto:abusecomplaints@markmonitor.com)",  
"[dns-admin@google.com](mailto:dns-admin@google.com)"  
]  
从结果可以看出该域名属于Google，实际上确实如此。该域名是用于Google App Engine服务的。当我们爬取该域名是就需要十分小心，因为Google经常会阻断网络爬虫，尽管实际上其自身就是一个网络爬虫业务。

**4.爬虫陷阱**

一些网站会动态生成域名内容，这样就会出现无限多的网页。比如，网站有一个在线日历功能，提供了可以访问下个月和下一年的链接，那么下个月的页面中同样会包含访问再下个月的链接，这样页面就会无止境地链接下去。  
修改seen变量

**5.抓取（scraping）**

我们需要让这个爬虫从每个网页中抽取一些数据，然后实现某些事情，这种做法被称为抓取。

**6.分析网页**

浏览器中右键单击选择View page source选项，获取网页的源代码。

**7.抓取网页的方法**

**Beautiful Soup该模块可以解析网页，并提供定位内容的便捷接口。  
pip install beautifulsoup4**

from bs4 import BeautifulSoup  
broken\_html ='

Area

Population

'  
#parse the HTML  
soup =BeautifulSoup(broken\_html,'html.parser')  
fixed\_html=soup.prettify()  
print fixed\_html

=============== RESTART: F:\python1\beautifulsoup1.py===========

* Area
* Population

---------------------------------------------------------------------从上面的执行结果可以看出，BeautifulSoup能够正确解析缺失的引号并闭合标签，此外还添加了和标签使其成为完整的HTML文档。现在使用find()和find\_all()方法来定位我们需要的元素。

参考：  
<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

还有Lxml是基于libxml2这一SML解析库的Python封装。该模块使用C语言编写，解析速度比BeautifulSoup更快，不过按照过程也更为复杂。  
参考：  
<http://lxml.de/installation.html>  
也可以使用正则表达式；